VISIT…

LANZAROTE
Caliente.com

ПРИРОДА

117049 Москва, Мароновский пер., д. 26
тел. : [095] 238-24-56; e-mail: byalko@landau.ac.ru

№ 6 - 1999 г.

В.П. Ильин

Линейная алгебра: от Гаусса до суперкомпьютеров будущего

© Природа

Использование или распространение этого материала в коммерческих целях возможно лишь с разрешения редакции



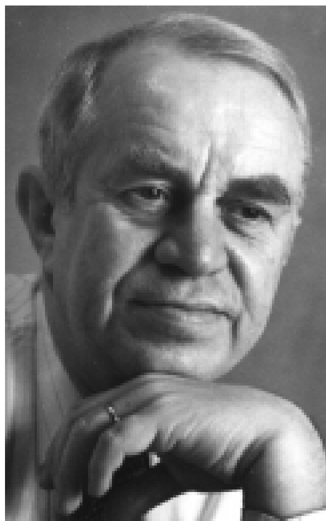
Образовательный сетевой выпуск
VIVOS VOCO! - ЗОВУ ЖИВЫХ!
http://www.techno.ru/vivovoco



Линейная алгебра: от Гаусса до суперкомпьютеров будущего

В.П.Ильин

...Поверил
Я алгеброй гармонию
А.С.Пушкин



Валерий Павлович Ильин, доктор физико-математических наук, профессор, главный научный сотрудник Института вычислительной математики и математической геофизки СО РАН. Область научных интересов — вычислительная математика, информатика, математическое моделирование. Автор более 250 научных статей и девяти монографий (часть из них написана в соавторстве).

ВВЕДЕНИЕ: ИСТОРИЯ, ПРЕДПОСЫЛКИ И ПРОБЛЕМЫ

Задача численного решения систем линейных алгебраических уравнений (СЛАУ) имеет незапамятную историю. Классический метод исключения, активно развиваемый и изучаемый даже в наши дни, был открыт К.Гауссом в 1849 г. Однако еще за 2000 лет до этого в Древнем Китае изданы "Девять книг о математическом искусстве", где этот алгоритм уже изложен в характерной для своего времени "натуральной" форме, но фактически с использованием матричных преобразований.

Вплоть до начала XX в. алгебра оставалась "наукой о решении уравнений", после чего произошло ее разделение на высшую алгебру (операции с абстрактными объектами различной природы) и линейную, основа которой — матричное исчисление.

Становление современных вычислительных методов линейной алгебры можно считать состоявшимся после выхода в 1960 г. книги Д.К. и В.Н.Фаддеевых с одноименным названием, по которой училось не одно поколение российских и зарубежных математиков. Необходимо подчеркнуть, что актуальные проблемы вычислительной алгебры имеют фундаментальный характер не только потому, что текущая компьютеризация различных областей знаний в значительной степени сводится к векторно-матричным процедурам. Изучение матриц, являющихся операторами в простейших конечномерных пространствах, позволя-

© В.П.Ильин

ет обнаружить наиболее глубокие и тонкие свойства математических объектов, имеющих свое значение для функционального анализа, теории аппроксимации, дифференциальных уравнений и т.д.

Последние десятилетия ознаменованы бурным развитием численных методов, нашедших отражение в книгах Г.И.Марчука, А.А.Самарского, С.К.Годунова, Дж.Голуба, О.Аксельсона и др. В рамках итерационных методов решения СЛАУ значительные продвижения достигнуты в таких современных направлениях, как алгоритмы сопряженных направлений, последовательной и симметричной верхней релаксации, неявные алгоритмы переменных направлений и методы неполной факторизации.

Несмотря на имеющиеся в этой области теоретические достижения, а также на бурный рост мировых вычислительных мощностей, проблема конструирования и исследования новых быстрых алгоритмов решения СЛАУ не теряет своей актуальности, и поток публикаций по этой тематике только увеличивается. Дело объясняется тем, что возникающие практические задачи наряду с усложнением вычислительных процедур требуют от методов решения все большей точности и надежности. Как говорится, аппетит приходит во время еды, и с этим связан тот уникальный феномен, что, невзирая на появление все новых поколений ЭВМ, за последние 50 лет решение "больших" (на текущий момент) задач всегда занимало примерно одинаковое астрономическое время.

Образно говоря, можно сформулировать теорему о существовании перманентного дефицита машинных ресурсов, стимулирующего как дальнейшее наращивание компьютерных мощностей, так и развитие новых супералгоритмов.

Главный источник сверхбольших СЛАУ — сеточные методы (конечных разностей, конечных объемов и конечных элементов) решения многомерных краевых задач, возникающих при моделировании процессов или явлений. Для оценки потребностей вычислительных ресурсов рассмотрим сеточную трехмерную расчетную область в форме параллелепипеда, образованную координатными линиями

$$x=x_i,\ i=1,\dots,I;\ y=y_j,\ j=1,\dots,J;\ z=z_k,\ k=1,\dots,K.$$

Если мы решаем одно дифференциальное уравнение второго порядка эллиптического типа (например, уравнение стационарной диффузии или теплопроводности), то в результате его простейшей аппроксимации для каждого узла (i,j,k) получается уравнение вида

$$\begin{aligned}(Au)_{i,j,k} &= p_0u_{i,j,k} - p_1u_{i-1,j,k} - \\ &- p_2u_{i,j-1,k} - p_3u_{i+1,j,k} - p_4u_{i,j+1,k} - \\ &- p_5u_{i,j,k-1} - p_6u_{i,j,k+1} = f_{i,j,k},\end{aligned} \qquad (1)$$

где коэффициенты p_k выражаются через шаги сетки и параметры исходной задачи. Такой системе уравнений отвечает квадратная матрица A порядка IJK с числом ненулевых элементов на каждой строке не более 7.

Пусть ради простоты $I=J=K>>1$. Тогда для реализации метода исключения Гаусса требуется выполнить примерно $Q=K^7$ арифметических операций и запомнить $P=K^5$ промежуточных чисел. На сегодня задаче средней сложности соответствует $K=100$, что приводит к значениям $Q=10^{14}$ и $P=10^{10}$. На обиходном языке это означает около трех часов работы компьютера быстродействием 10 гигафлоп, т.е. 10^{10} "флоп" — арифметических операций над "нормальными" вещественными числами в секунду, при наличии оперативной памяти около 10 гигабайт. Это намного превосходит ресурсы персональных компьютеров и рабочих станций, но на мировом рынке такая ЭВМ — среднего класса. Однако если перейти к "большой" задаче с $K=1000$, то она уже оказывается крепким орешком даже для рекордного суперкомпьютера со скоростью в один терафлоп (10^{12}) и памятью в один терабайт.

Двумерные задачи гораздо легче. Однако и они требуют огромных ресурсов, если нужно находить оптимальные параметры искомых решений (чтобы удовлетворить поставленным ограничениям и выбранным критериям качества), рассчитывать длительные по времени

или, наоборот, быстроменяющиеся процессы, учитывать сильнонелинейные эффекты. Это приводит к многократному решению промежуточных линейных систем. Важно отметить, что это — наиболее трудоемкая часть крупномасштабного вычислительного эксперимента, так как остальные расчетные этапы требуют объема арифметических действий, пропорционального общему количеству узлов сетки.

Кардинальное средство повышения производительности компьютеров (по профессиональным прогнозам — до петафлопного быстродействия, т.е. до 10^{15} операций в секунду к 2015 г.) — создание многопроцессорных вычислительных систем (до 10^3 или 10^6 процессоров). Отсюда одной из первых по актуальности проблем в вычислительной алгебре становится распараллеливание алгоритмов и их отображение на архитектуру многопроцессорных ЭВМ. Насущность этого вопроса, в том числе при решении СЛАУ, проистекает из того общеизвестного факта, что эффективность загрузки всех устройств — это больное место различных суперкомпьютеров, и реализация алгоритмов без учета, например, коммуникационных потерь может снизить фактическую производительность многопроцессорной системы до нескольких процентов от ее паспортных данных.

Один из самых эффективных методов решения СЛАУ высокого порядка базируется на итерационных алгоритмах неполной факторизации, получивших сейчас самое широкое распространение в мировой вычислительной практике[1]. Данная статья представляет последние результаты автора, касающиеся новых, перспективных подходов: обобщенного принципа компенсации в приближенной матричной факторизации, адаптивных итерационных методов с оптимальными упорядоченностями и, наконец, проблемы распараллеливания алгоритмов на основе декомпозиции областей.

[1] Il'in V.P. Iterative incomplete factorization methods. Singapore, 1992; Ильин В.П. Методы неполной факторизации для решения алгебраических систем. М., 1995.

СОВСЕМ НЕ АЛГЕБРАИЧЕСКАЯ, НО ОЧЕНЬ ПЛОДОТВОРНАЯ ИДЕЯ

Идея метода неполной факторизации для решения алгебраической системы

$$Au=f \qquad (2)$$

заключается в построении такой матрицы B, называемой предобусловливающей, которая легко бы обращалась и была бы близкой к исходной матрице A в том смысле, что собственные числа матрицы-произведения $B^{-1}A$ не сильно различаются между собой. Тогда последовательные приближения u^n строятся с помощью итерационного процесса, который в канонической записи имеет следующий вид:

$$B(u^n - u^{n-1}) = \omega_n(f - Au^{n-1}). \qquad (3)$$

Здесь ω_n — вычисляемые параметры, а число итераций $n(\varepsilon)$ для достижения требуемой точности $\varepsilon \ll 1$ зависит от отношения

$$\kappa = \max|\lambda|/\min|\lambda|,$$

называемого числом обусловленности матрицы $B^{-1}A$. Эта величина κ оказывается магическим числом в линейной алгебре, характеризуя главные вычислительные качества любой матрицы, в том числе чувствительность к погрешностям округлений.

Вводя обозначения $v^n = u - u^n$, $r^n = f - Au^n$ для векторов коррекции и невязки, соотношение (3) можно разбить на два:

$$Bv^n = r^{n-1}, \quad u^n = u^{n-1} + \omega_n v^n,$$

из которых видно, что наиболее трудоемкая процедура на каждой итерации — решение вспомогательной системы с матрицей B (конечно, если последняя не слишком тривиальна).

Первооткрыватель методов неполной факторизации Н.И.Булеев предложил (середина 50-х годов, закрытый ядерный центр в г.Обнинске) искать B как

приближенную (неполную) факторизацию исходной матрицы A:

$$B = LU \cong A, \quad (4)$$

где L, U — легко вычисляемые (намного проще, чем в точной факторизации) и легко обратимые нижняя и верхняя треугольные матрицы.

Гениальное все просто: помимо конструктивного построения множителей L, U, Булеев предложил в качестве критерия близости матриц требование

$$Be = Ae, \quad (5)$$

где e — вектор с постоянными (единичными) компонентами; он назвал его принципом компенсации. Эта совсем не алгебраическая идея исходила из тех сермяжных соображений, что в задачах матфизики решения в основном гладкие, а тогда при условии (5) на каждой итерации "лишние" члены (обусловленные неточностью факторизации (4)) будут взаимно компенсироваться.

Новые методы на многих практических задачах обнаруживали потрясающее быстродействие, но временами не все было гладко. Это говорило о необходимости более глубокого изучения нового класса алгоритмов, теории которых тогда не было.

Но, как известно, нет пророка в своем отечестве — ведущие специалисты в СССР идею неполной факторизации не подхватили, продолжая увлеченно развивать другие (пришедшие из-за рубежа) модные направления.

На Западе же методы Булеева были несколько раз переоткрыты и стали активно усовершенствоваться. В частности, эвристическое условие (5), которое на алгебраическом языке не что иное, как принцип согласования строчных сумм матриц B и A, было объяснено и эффективно использовано в сочетании с ускорением итераций универсальными методами сопряженных градиентов. Поразительное совпадение, но практически одновременно (в 1983 г.) и независимо в работах автора с О.Аксельсоном (Голландия) и Дж.Голуба (США) с коллегами были предложены неявные (блочные) методы неполной факторизации, в которых матричные множители L, U — суть блочно-треугольные матрицы, причем диагональные блоки — ленточные матрицы, ненулевые элементы которых сосредоточены только в окрестности ленты заданной ширины d около главной диагонали. Такие алгоритмы позволили уточнить саму приближенную факторизацию и значительно сократить число итераций. Однако здесь возникла новая нетрадиционная задача вычислительной алгебры: найти ленточную часть матрицы, обратной к ленточной же матрице. Для этой задачи удалось найти изящное решение (естественно, трудоемкость процедуры увеличивается с ростом ширины ленты d).

В этом месте заложена принципиальная методологическая проблема, ожидающая до сих пор своего решения. Пусть, например, C — невырожденная матрица. Если мы вычисляем ленточную часть ширины $d>1$ от обратной матрицы — обозначим ее через $(C^{-1})^{(d)}$, — то находим в определенном смысле приближение к обратной матрице C^{-1} (которая в общем случае плотная, т.е. не содержит нулей). Действительно, если d увеличивается вплоть до порядка матрицы N, то $(C^{-1})^{(d)} \to C^{-1}$. При $d=N$ матричная аппроксимация становится точной и метод решения СЛАУ из итерационного превращается в прямой, дающий точное решение за конечное число действий. Фактически в этом случае метод приближенной факторизации переходит в блочный метод Гаусса, который для сеточных систем уравнений слишком дорог. Здесь и возникает вопрос: какова оптимальная ширина $1 \le d \le N$ ленточной аппроксимации обратной матрицы $(C^{-1})^{(d)}$, чтобы обеспечить итоговую эффективность итерационного метода? Или, другими словами, достичь заданной точности за минимальное число арифметических действий? Понятно, что лучшее — враг хорошего, и в полном объеме решение проблемы оптимизации алгоритмов — это недостижимая голубая мечта, к которой надо стремиться, но только дробными шагами, рас-

ширяя жизненное пространство конкретными результатами для отдельных типов систем.

ОТ ЛИНЕЙНОЙ НЕЗАВИСИМОСТИ ВЕКТОРОВ — К НЕЗАВИСИМОСТИ СИЛЬНОЙ

Одна из генеральных линий развития итерационных методов заключается в обобщении булеевского принципа компенсации (5) на основе перехода к системе так называемых пробных векторов y_q, для которых должны выполняться равенства

$$By_q = Ay_q, \quad q = 1, \ldots, m. \qquad (6)$$

Здесь результаты получены для уникального класса вещественных блочно-трехдиагональных матриц — стилтьесовых, которые, с одной стороны, лежат в основе решения широкого класса практических задач, а с другой — обладают свойствами, полезными при исследовании основанных на них алгоритмов. Главные из этих свойств — симметричность, положительность собственных чисел и монотонность (обратная матрица A^{-1} имеет неотрицательные элементы).

Изначальный вопрос здесь: для каких наборов пробных векторов y_q вообще разрешима задача о вычислении предобусловливающей матрицы B рассматриваемого факторизованного вида? Изучение проблемы привело автора к новому понятию в линейной алгебре — сильной независимости векторов. Если в матрице размером $m \times N$, составленной из m векторов y_q порядка $N > m$, каждый минор m-го порядка ненулевой, то такие векторы назовем сильно независимыми. Подчеркнем, что обычной, линейной, независимости векторов (когда хотя бы один минор ненулевой) недостаточно для разрешимости задачи (6).

Далее следуют обычные для вычислительной математики вопросы, на которые удалось получить первые обнадеживающие ответы: как конструктивно построить экономичный алгоритм по условию (6)? Какова его чувствительность и устойчивость к погрешностям округлений машинных операций? При каких пробных векторах предобусловливающая матрица B будет симметричной и положительно определенной (что необходимо, если требуется исследовать сами методы неполной факторизации)? Сколько и каких пробных векторов надо брать, чтобы эффективность алгоритма была по возможности высокой (мы сознательно не употребляем слова "наивысшей")? Какова будет скорость сходимости итераций? Поиск решений — новое направление, выходящее за рамки теории только итерационных алгоритмов.

Уравнения (6) можно интерпретировать как обратную обобщенную задачу на собственные значения: для заданных собственных векторов y_q и матрицы A найти матрицу B (среди рассматриваемого класса ленточных матриц), которая соответствует единичному обобщенному собственному числу. Такая нетрадиционная задача линейной алгебры, для которой уже найдены некоторые частные решения, наверняка найдет свою нишу в различных приложениях.

Другой аспект принципа обобщенной компенсации имеет прямое отношение к теории приближений. Если начальную ошибку итераций представить в виде линейной комбинации пробных векторов:

$$u - u^0 = c_1 y_1 + \ldots + c_m y_m, \qquad (7)$$

то первое итерационное приближение u^1 в (3) при $\omega_1 = 1$ уже будет точным решением. Справедливо и более сильное утверждение: среди систем сильно независимых пробных векторов оптимальна та, которая наилучшим образом приближает искомое решение u (при $u^0 = 0$). А поиск таких приближений уже можно проводить с помощью априорных оценок для дифференциальных краевых задач.

И наконец, третий важный момент, чисто алгебраический: соотношение (7) позволяет исследовать итерационные процессы вида (3) в подпространстве, натянутом на векторы $y_1, \ldots y_m$, — что тоже является актуальной задачей серьезного теоретического и практического значения.

УХУДШЕНИЕ АЛГЕБРАИЧЕСКИХ СВОЙСТВ СИСТЕМЫ МОЖНО ОБРАТИТЬ ВО БЛАГО

Одна из общих современных проблем вычислительной математики — построение адаптивных алгоритмов, подстраивающихся под свойства решаемой задачи и повышающих при этом свою эффективность на основе анализа априорной и/или апостериорной информации. Наибольшее значение такое качество имеет при решении трудоемких задач. В математической физике к таковым относятся "жесткие" уравнения с очень резко меняющимися свойствами решений. В качестве примера можно привести анизотропное уравнение стационарной диффузии (или теплопроводности)

$$-\frac{\partial}{\partial x} a \frac{\partial u}{\partial x} - \frac{\partial}{\partial y} b \frac{\partial u}{\partial y} = f(x, y), \qquad (8)$$

в котором коэффициенты a,b могут сильно изменяться (даже на несколько порядков величины) и иметь в том числе большие разрывы. У краевых задач подобного рода очень "плохие" решения, которые порождают главные проблемы с качеством аппроксимации, правда, здесь они волнуют нас в меньшей степени. Другая особенность такой задачи заключается в катастрофически плохой обусловленности алгебраической системы, получаемой после дискретизации. Помимо потери вычислительной устойчивости, это приводит, как правило, к значительному увеличению количества итераций $n(\varepsilon)$. В работах автора с Б.Келлогом (США) на примере простейшего итерационного метода Зейделя, по-видимому, впервые было замечено, что если вычисления каждого следующего приближения проводить последовательно по узлам, упорядоченным с учетом направления изменений коэффициентов исходных уравнений, то наличие их больших градиентов можно обратить в свою же пользу. Получается парадоксальный на первый взгляд феномен: ухудшение алгебраических свойств системы приводит к улучшению итерационного алгоритма, если он нужным образом адаптирован.

Этот результат удалось перенести и на такие "идеологически" сложные алгоритмы, как методы неполной факторизации. Доказан принципиальный факт ускорения сходимости адаптивных алгоритмов при монотонном изменении коэффициентов исходной системы. Выведенные теоретические оценки наглядно подтверждаются численными экспериментами. В двух словах, полученные утверждения можно объяснить следующим образом. При адаптивном построении предобусловливающая матрица B "наследует" и компенсирует плохие свойства исходной матрицы A, в результате чего качество итогового алгоритма только улучшается.

Аналогичный эффект наблюдается и в случае очень сильной анизотропии (например, $a \gg b$), даже если коэффициенты постоянны. Хотя при этом обусловленность исходной системы тоже ухудшается, в определенном смысле задача приближается к одномерной, и при соответствующем блочном упорядочении неизвестных итерационный алгоритм сводится к безытерационному.

Важно отметить, что адаптивные качества итерационных алгоритмов определяются непосредственно свойствами алгебраической системы. А ее элементы зависят как от поведения коэффициентов дифференциальной задачи, так и от шагов сетки. Поэтому рассматриваемые вычислительные явления могут быть вызваны не "физическими", а чисто дискретизационными эффектами: если сетка слишком неравномерная (а при сильной разномасштабности деталей расчетной области шаги могут отличаться в сотни и тысячи раз), то обусловленность матрицы A сильно ухудшается, но путем соответствующего упорядочения узлов можно сходимость итераций, наоборот, улучшить.

Исследование и построение таких адаптивных алгоритмов сильно усложняются, если коэффициенты уравнений описываются немонотонными функциями с многочисленными экстремальными точками. Имеющийся опыт по исследо-

ванию простейшего метода Зейделя в таких ситуациях позволяет надеяться, что проблемы здесь чисто технического характера: во-первых, в автоматизации определения “хитрых” последовательностей с монотонными свойствами и, во-вторых, в обнаружении необходимых свойств матриц с более сложными блочными структурами.

Отметим, что рассмотренный адаптивный подход выпадает из большой серии работ известных авторов. Эти работы посвящены обоснованию различных итерационных алгоритмов, у которых скорость сходимости не зависит от величины разброса исходных коэффициентов. Однако противоречия здесь на самом деле нет, так как построение адаптивных алгоритмов — это использование дополнительной информации о внутренних свойствах задачи.

УВЯЗАТЬ НЕСОВМЕСТИМОЕ

Основная методология реализации вычислительных процессов на многопроцессорных компьютерных системах — это выявление таких счетных фрагментов алгоритма, которые можно выполнить одновременно на различных арифметических устройствах при минимальных информационных обменах между ними. Это объясняется чисто техническими деталями: время обмена массивов из m чисел между двумя соседними процессорами, которые имеют непосредственную физическую связь, можно выразить как $T_c = \tau_0 + m\tau_c$, где τ_c есть длительность передачи одного числа, а τ_0 — время стартовой коммуникационной задержки. Если через τ_a обозначить усредненное время выполнения одной арифметической операции, то обычно имеем $\tau_0 \gg \tau_c \gg \tau_a$, где конкретные соотношения времен могут меняться, но в главном эти неравенства сохраняются.

Если не вдаваться в глубокие тонкости отображения алгоритмов на архитектуру вычислительной системы, эффективность распараллеливания можно выразить количественно двумя основными характеристиками — коэффициентами ускорения и загрузки процессоров:

$$R_p = \frac{T_1}{T_p}, \quad E_p = \frac{R_p}{p},$$

где T_1, T_p — интервалы времени выполнения алгоритма (или задачи) на одном и p процессорах, причем T_p оценивается с учетом коммуникационных потерь.

Надо подчеркнуть, что ускорение алгоритма и задачи могут быть разными, поскольку для решения последней может оказаться целесообразным выбрать метод, который требует большего объема вычислений, но лучше распараллеливается.

Если рассматривать проблему распараллеливания решений многомерных краевых задач в целом, то решение СЛАУ также один из главных этапов, так как другие расчетные стадии — дискретизация области, аппроксимация дифференциальных уравнений и обработка результатов — распараллеливаются гораздо легче. Тривиально распараллеливаются и простейшие итерационные алгоритмы, но они имеют слишком плохую скорость сходимости. Трудности в реализации наиболее быстрых методов заключаются в том, что они основаны на выполнении различных рекуррентных соотношений, являющихся сугубо последовательными операциями.

Наиболее перспективный подход к распараллеливанию краевых задач заключается в декомпозиции расчетных областей на подобласти с возможно одинаковым количеством узлов в каждой. При этом общая задача разбивается на подзадачи, каждая из которых решается на своем процессоре, с организацией периодических информационных интерфейсов между ними. Если решается СЛАУ, то в каждой подобласти итерируется соответствующая подсистема, что приводит в итоге к двойному итерационному процессу.

На примере декомпозиции трехмерных задач можно проследить ряд важных методологических вопросов, тесно связанных с чисто архитектурными компьютерными аспектами. Вычислительная сеть может представлять собой одномерную сетку, двумерную прямо-

угольную или трехмерную параллелепипедоидальную, когда каждый процессор физически связан соответственно с двумя, четырьмя или шестью топологически соседними процессорами (мы не будем рассматривать другие экзотические и более дорогие сети типа гиперкуб и "каждый с каждым"). Теперь, если у нас есть трехмерная расчетная сеточная область с числом узлов $N=K^3$, много большим количества процессоров p, то как лучше ее разбить на подобласти с точки зрения оптимальности распараллеливания?

Ответ зависит от соотношения времен T_a и T_c, затрачиваемых такими вычислительными системами соответственно на выполнение арифметических операций и обмен данными между процессорами. А это связано с чисто геометрическими факторами. Величина T_a при вычислении одной итерации пропорциональна общему числу узлов подобласти K^3/p, т.е. ее объему, а значение T_c пропорционально количеству граничных узлов, смежных с соседними подобластями. При использовании одномерной вычислительной сети область разбивается на "пластины" с площадью двух смежных граней (числом граничных узлов), равной $2K^2$. В случае "двумерной" декомпозиции — на "столбики", площадь смежных граней которых $4K^2/\sqrt{p}$ (здесь предполагается компьютерная сеть с числом процессоров $\sqrt{p}$ в каждом из двух измерений). Наконец, при определении подобластей в виде "кубиков с длиной ребра $Kp^{-1/3}$" величина T_c пропорциональна $6K^2p^{-2/3}$. Таким образом, при $p \gg 1$ с точки зрения коммуникационных потерь "одномерная" декомпозиция оказывается наихудшей, а трехмерная — наилучшей.

Разумеется, здесь "за кадром" остается много других вопросов и возможностей: совмещение во времени обменов процессора с несколькими соседними, одновременное выполнение обменов и вычислений, использование сверхбыстрой кэш-памяти, конвейеризация арифметических операций в отдельном процессоре, если он содержит, например, несколько сумматоров и множительных устройств.

Эффективная реализация параллельных алгоритмов предполагает грамотное использование архитектуры компьютера и систем программирования, что значительно повышает планку требований к квалификации прикладного математика-программиста. С прагматической точки зрения это, конечно, крупный недостаток, и на преодоление таких накладных расходов ориентированы существующие и развиваемые средства автоматизации распараллеливания, которые в идеале должны полностью скрыть от пользователя внутреннюю технологическую "кухню", при сохранении высоких коэффициентов R_p и E_p.

ЗАКЛЮЧЕНИЕ: КАМО ГРЯДЕШИ?

Разумеется, мы не могли исчерпать обилия интереснейших вопросов даже в рамках методов неполной факторизации. В первую очередь — проблемы распространения полученных результатов на более общие задачи: нерегулярные сетки, матрицы с менее сильными свойствами и усложненными структурами, краевые задачи для систем дифференциальных уравнений и т.д.

Кроме того, обсуждавшиеся алгоритмы развиваются одновременно с другими перспективными направлениями, среди которых — многосеточные методы, альтернирующие операторные принципы, многоуровневые иерархические итерации. И здесь, кроме здоровой конкуренции, — неизбежно взаимное обогащение с проникновением идей и конструктивных подходов. При существующем росте научных школ и бурном потоке исследований трудно загадывать их тенденции даже на не очень далекое будущее, но текущий период можно без преувеличения назвать золотым веком вычислительной алгебры.

Статья написана по результатам исследований, поддержанных грантом РФФИ 95-01-01770 "Методы решения систем линейных алгебраических уравнений высокого порядка с разреженными матрицами".